Exhibit 3

Copy of an Internal, Confidential, and Non-Publicly Disclosed Document Written by Inventor Ikko Fushiki, entitled, "Extension of sRGB color and GDI+"

(Dates Redacted)
(4 pages)

Extension of sRGB color and GDI+

Ikko Fushiki

Introduction

Color monitors, color scanners, and color printers are becoming the common equipments in companies, home offices, and homes. The color management among the different devices is important ever for the desktop publishing. The color standard sRGB was introduced to handle most of uses for non-publishers. Unlike CIE Lab or other color standard, sRGB has easier conversion rules for monitors and also it can be used as a color reference for the other types of devices. However, the conventional publishers use Pantone colors and CMYK values as a reference for printing. The demands for using those colors are still high for the high-end publishing and there is the criticism of sRGB in its narrow range of gamut. We must respond to those needs.

GDI+ has various features of gradient fills. When colors c1 and c2 are averaged, we expect to have the perceptually averaged color. However, sRGB has non-linearity produces the problems. Hence simple averaging of sRGB values does not produce the desired visual effects. Also GDI+ uses alpha channel for translucency. When the translucency is 50 %, simple multiplication of 0.5 to the sRGB value does not produce the 50 % translucent effects. We must linearize the sRGB value before we apply those operations to the color. When the linearized information is stored in 8 bit in each channel, we lose the accuracy of the original data in the lower values. Some contouring artifacts will appear and the resultant image does not look smooth.

We would like to solve the problems of narrow gamut and non-linearity of sRGB by proposing XsRGB format.

Extension of sRGB and Gamut

Figure 1. Extended sRGB space in CIE Diagram

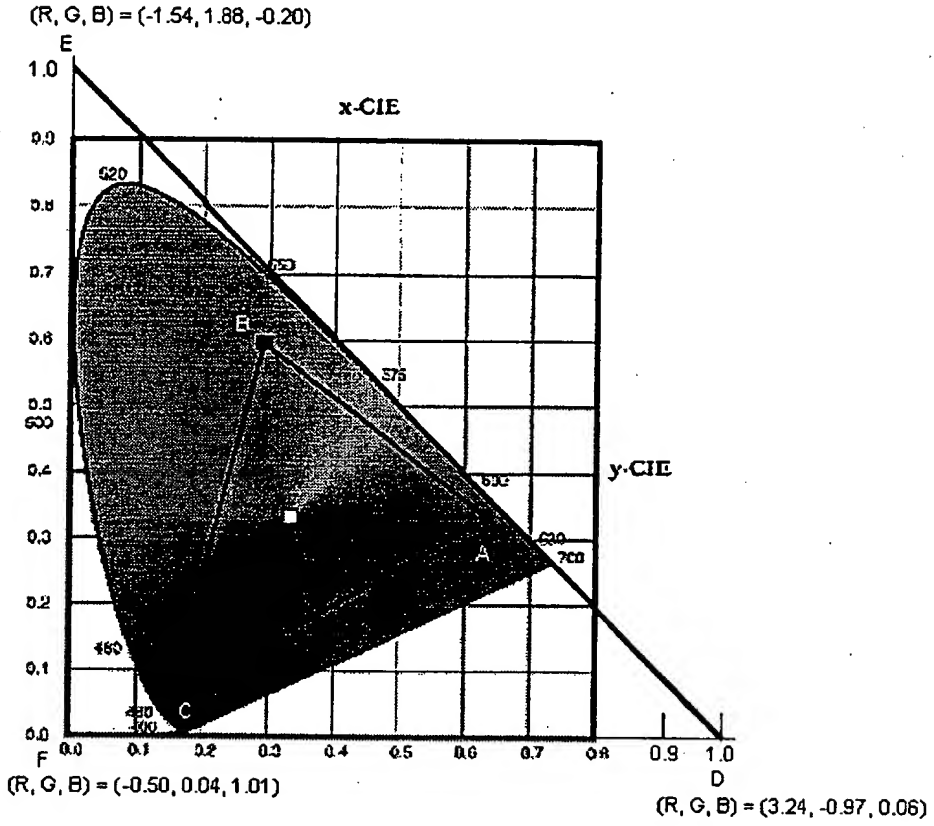

Figure 1 is a CIE chromaticity diagram. The points A, B, and C represent the red, green, and blue points in sRGB. Some printers go beyond sRGB gamut in cyan colors. One approach to extend the gamut is to use a laser display device as a color reference. It will cover most of the range. However, we need extra calculation to convert the laser device colors to the ordinary CRT colors (sRGB). Instead we can extend the range of sRGB beyond 1 and below 0. The ideal device can cover a triangle DEF in Figure 1. This corresponds to the linear sRGB values of (3.24, – 0.97, 0.06), (– 1.54, 1.88, – 0.20), and (– 0.50, 0.04, 1.01). If we want to express those values in 16-bit number, we must use 1.2.13 (1 bit for sign, 2 bit for integer, and 13 bit for decimals) format. This will cover each component within – 4 to 4.

Keeping the non-linear sRGB in 16-bit is problem. First of all, it is not clear how sRGB profile will be extended beyond 1 and below 0. Second, if we extend sRGB smoothly

(but linearly) beyond 1, the red value at point D reaches nearly 6. Hence, 1.2.13 format is not enough for non-linear sRGB.

I suggest that we always use the linear sRGB in 16-bit format. Each component can have 8192 levels of shades. The linear and non-linear conversion is accurate in that level. When we extend the range of RGB beyond 1 and below 0, the issue of the conversion between XsRGB and sRGB appear. We must discuss the conversion strategy and must have the standard methods among GDI+ team and ICM venders.

Color Support

GDI+'s architecture is based on sRGB and will be extended to XsRGB if the agreements are met between Microsoft GDI+ group and ICM venders. In addition to the color space, GDI+ uses alpha channel. Hence, GDI+ will have (X)sRGB + alpha for the basic color components. We interpret alpha as the coverage value of the pixel and its value is linearly scaled. Since ICM does not deal with alpha values, we do not modify alpha in any color format. "Color Support" here means conversions between (X)sRGB and different color format like CMYK, etc.

The first version of GDI+ APIs will be based on (X)sRGB. The future version will fully support ICM. So it is important to discuss the strategy of color conversion and color profile with venders. For a preliminary support, GDI+ may supply color conversion routines among various color spaces.

This is different from Java 2D. There are Color class and ColorSpace class in Java 2D. Color contains ColorSpace and Color itself could be CMYK or CIE Lab according to its ColorSpace. Each drawing primitives must be able to handle different color space. If we use this approach, GDI+ implementation will be very complicated. We must receive the feedback from printer venders if GDI+ need to have similar approach.

There are other issues in color space. One is the named color space. Offset printing venders may want to have Pantone color names. This can be integrated into GDI+ itself. For Pantone color images, we can introduce the index bitmaps that can contain the Pantone color names in addition to its (X)sRGB values or CMYK values. Image importer and image exporter can call a color converter if necessary. If a printer recognizes the color names, it can use them. If not, the pixel values in the color table is used in printing.

ICM and GDI+

ICM can convert one color space to another. The color spaces that ICM can transform are Gray, RGB, CMYK, XYZ, Yxy, Lab, etc. Other features of ICM is that it can read and create the color profiles. When we extend sRGB to XsRGB, the venders must agree

the XsRGB format and produce the profiles. The profile vender do not need to worry about the artificial gamut limitations imposed by sRGB.

If we leverage ICM codes or create wrapper to create ColorConverter class, GDI+ will satisfy the needs of color publishers as well as desktop publishers. The extension of ICM and its integration to GDI+ will bring the wide acceptance of GDI+ graphics among the color publishing industry.